## AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A computer-implemented method, said method comprising:
providing a graphical user interface for defining at least one function to be used in a graphical representation of a finite state machine, where the graphical representation is an executable model of the finite state machine;

representing the at least one function graphically[[;]], wherein the function that is represented graphically is a function defined in a graphical language; and

calling the function that is represented graphically from within the graphical representation of the finite state machine.

2. (Previously Presented) The method of claim 1 wherein defining the at least one function further comprises using a function block.

3. (Previously Presented) The method of claim 2 wherein defining the at least one function further comprises using a function prototype.

4. (Original) The method of claim 1 wherein the defining step further comprises using a function flow diagram.

5. (Currently Amended) ~~The method of claim 1~~ A computer-implemented method, said method comprising:

providing a graphical user interface for defining at least one function to be used in a graphical representation of a finite state machine, where the graphical representation is an executable model of the finite state machine;

representing the at least one graphically, wherein the function is represented graphically as a diagram comprising graphical elements; and

calling the function that is represented graphically from within the graphical representation of the finite state machine.

6. (Previously Presented) The method of claim 1 further comprising modifying the at least one function through graphical diagramming.

7. – 11. (Cancelled)

12. (Currently Amended) A computer program product, stored in a computer readable medium, comprising instructions to cause a computer to:

receive user input defining at least one graphical function for use in a finite state machine; and

use the at least one graphical function in a simulation of a system represented by the finite state machine, wherein the instructions to use the at least one graphical function further comprise instructions to call the at least one graphical function from at least one state or transition in the finite state machine.

13. (Original) The computer program product of claim 12 wherein the user input defining the at least one graphical function is entered into a function block.

14. (Original) The computer program product of claim 12 wherein the user input defining the at least one graphical function includes a function prototype.

15. (Original) The computer program product of claim 12 wherein the user input comprises a function flow diagram.

16. (Previously Presented) The computer program product of claim 15 wherein the function flow diagram is comprised of graphical elements.

17. (Currently Amended) A system for modeling finite state machine, said system comprising:

a computer comprising a graphical user interface, memory, storage, and at least one input device;

means to receive user input to define at least one graphical function;

means to represent the graphical function as an executable ~~a~~state flow diagram; and

means to ~~use~~ call the graphical function from at least one finite state machine in a simulation of the at least one finite state machine.

18. (Previously Presented) The system of claim 17 wherein the user input to define the at least one graphical function is entered into a function block.

19. (Canceled)

20. (Previously Presented) The system of claim 17 wherein the user input to define the at least one graphical function includes a function prototype.

21. (Original) The system of claim 17 wherein the user input comprises a function flow diagram.

22. (Previously Presented) The system of claim 21 wherein the function flow diagram is comprised of graphical elements.

23. (Previously Presented) The system of claim 21 further comprising means for hiding the display of the function flow diagram based upon user input.

24. (Currently Amended) A method of operating a data processing system having a graphical user interface, said method comprising:

creating a graphical representation of a finite state machine and a graphical representation of a function for use in the graphical representation of the finite state machine; and

simulating a system represented by the finite state machine, wherein the graphical representation is an executable model of the system; and

calling the function from the executable model of the system during the act of simulating the system represented by the finite state machine.

25. (Original) The method of claim 24 wherein the graphical representation of the function comprises a function prototype.

26. (Previously Presented) The method of claim 25 wherein the function prototype defines a textual format for invoking the function.

27. (Previously Presented) The method of claim 26 wherein the graphical representation of the finite state machine includes at least one invocation of the function using the defined textual format.

28. (Previously Presented) The method of claim 24 further comprising shadowing a function, wherein shadowing comprises using in a function invocation a function definition closest to a point of invocation of the function in a state diagram hierarchy.

29. (Previously Presented) The method of claim 24 wherein the function is exportable by a state chart and may be invoked anywhere in the finite state machine in which the chart appears, including other charts that define the finite state machine.

30. (Previously Presented) The method of claim 24 wherein simulating the system represented by the finite state machine further comprises computer code generation.

31. (Previously Presented) The method of claim 24, wherein the graphical representation of the function comprises a function prototype defining a textual format for invoking the function; and wherein the graphical representation of the finite state machine includes an invocation of the function using the defined textual format.

32. (Currently Amended) A computer readable medium having encoded thereon:
instructions for causing a computer system to receive through a graphical user interface a graphical representation of a finite state machine and a graphical representation of at least one function for use in the graphical representation of the finite state machine; and

instructions for simulating a system represented by the finite state machine, where the graphical representation is an executable model of the system; and

instructions for calling the function from at least one place in the executable model during the system simulation.

33. (Previously Presented) The computer readable medium of claim 32, wherein the graphical representation of the function comprises a function prototype defining a textual format for invoking the function; and wherein the graphical representation of the finite state machine includes an invocation of the function using the defined textual format.

34. (Currently Amended) In an electronic device, a method of graphically representing an event-driven system, said method comprising:

providing one or more block components representing one or more states in an executable model;

providing one or more transition components representing transitions between the one or more states; and

providing ~~a component representing~~ a function, said function comprising at least two graphical components and being referenced by at least one of the states or at least one of the transitions to call the function at the at least one of the states or the at least one of the transitions.

35. (Previously Presented) The method of claim 34, wherein the function accepts at least one argument and returns at least one result.

36. (Currently Amended) The method of claim 34, further comprising invoking the function at a second one of the one or more transition components or one or more block components.

37. (Previously Presented) The method of claim 34, further comprising specifying data properties of the function.

38. (Previously Presented) The method of claim 34, further comprising associating a data item with the function.

39. (Previously Presented) The method of claim 34, wherein the function comprises a graphical function.

40. (Previously Presented) The method of claim 34, wherein the function has a plurality of configurable properties.

41. (Previously Presented) The method of claim 34, wherein the function defines a textual format for invoking the function.

42. (Previously Presented) The method of claim 34, further comprising providing a shadowing function, wherein shadowing comprises using in a function invocation a function definition proximally closest to a point of invocation of the function in a state diagram hierarchy.

43. (Currently Amended) In a graphical representation environment, a system for graphically representing an event-driven system, said system comprising:

one or more block components representing one or more states in an executable model;

one or more transition components representing transitions between the one or more block components representing the states; and

a component representing a graphical function and referenced by at least one of the states or at least one of the transitions to call the function at one of the states or one of the transitions.

44. (Previously Presented) The system of claim 43, wherein the function accepts at least one argument and returns at least one result.

45. (Previously Presented) The system of claim 43, wherein at least a subset of the one or more block components representing the states and the one or more transition components can invoke the function.

46. (Previously Presented) The system of claim 43, further comprising means for specifying data properties of the function.

47. (Previously Presented) The system of claim 43, further comprising means for associating a data item with the function.

48. (Currently Amended) The system of claim 43, ~~wherein the function comprises a graphical function.~~ wherein the component representing the function is referenced by one more of: at least one of the states or at least one of the transitions.

49. (Previously Presented) The system of claim 43, wherein the function has a plurality of configurable properties.

50. (Previously Presented) The system of claim 43, wherein the function defines a textual format for invoking the function.

51. (Previously Presented) The system of claim 43, further comprising means for providing a shadowing function, wherein shadowing comprises using in a function invocation a function definition proximally closest to a point of invocation of the function in a state diagram hierarchy.

52. (Currently Amended) A medium for use in a graphical representation environment on an electronic device, the medium holding instructions executable using the electronic device for graphically representing an event-driven system, said instructions comprising instructions for:

providing one or more block components representing one or more states in an executable model;

providing one or more transition components representing transitions between the one or more block components representing the states; and

providing a block component representing a graphical function and referenced by at least one of the states or at least one of the transitions to call the function at one of the states or one of the transitions during execution of the event-driven system.

53. (Previously Presented) The medium of claim 52, wherein the function accepts at least one argument and returns at least one result.

54. (Previously Presented) The medium of claim 52, wherein the one or more transition components can invoke the function.

55. (Previously Presented) The medium of claim 52, further comprising instructions for accepting user input specifying data properties of the function.

56. (Previously Presented) The medium of claim 52, further comprising instructions for associating a data item with the function.

57. (Currently Amended) The medium of claim 52, wherein the function comprises two or more graphical ~~function~~ elements.

58. (Previously Presented) The medium of claim 52, wherein the function has a plurality of configurable properties.

59. (Previously Presented) The medium of claim 52, wherein the function defines a textual format for invoking the function.

60. (Previously Presented) The medium of claim 52 further comprising instructions for providing a shadowing function, wherein shadowing comprises using in a function invocation a function definition proximally closest to a point of invocation of the function in a state diagram hierarchy.

61. (Previously presented) A computer-implemented method for modeling a system using a graphical block diagram environment, said method comprising:

graphically representing a function for use in an executable model within the graphical block diagram environment; and

textually referencing the graphically represented function within the model to cause an invocation of the graphically represented function during execution of the model.

62. (Previously Presented) The computer-implemented method of claim 61, wherein the model is represented as a finite state machine.

63. (Previously Presented) The computer-implemented method of claim 62, wherein the finite state machine is a hierarchical finite state machine.

64. (Previously Presented) The computer-implemented method of claim 62 further comprising:

associating the graphically represented function with at least one state or transition within the finite state machine.

65. (Previously Presented) The computer-implemented method of claim 61, wherein the graphically represented function is represented as at least one of a finite state machine, a state flow diagram, a function flow diagram, and a graphical block diagram model.

66. (Currently Amended) A medium holding instructions executable using the electronic device for modeling a system using a graphical block diagram environment, said instructions comprising instructions for:

graphically ~~representing~~ defining a function for use in an executable model within the graphical block diagram environment; and

textually referencing the graphically represented function within the model to cause an invocation of the graphically represented function during execution of the model.

67. (Previously Presented) The medium of claim 66, wherein the model is represented as a finite state machine.

68. (Previously Presented) The medium of claim 67 further comprising instructions for: associating the graphically represented function with at least one state or transition within the finite state machine.

69. (Previously Presented) The medium of claim 66, wherein the graphically represented function is represented as at least one or a combination of: a finite state machine, a state flow diagram, a function flow diagram, and a graphical block diagram model.

70. (Currently Amended) A computer-implemented system for modeling using a graphical block diagram environment, said system comprising:

means for ~~graphically~~ representing a function defined graphically for use in an executable model within the graphical block diagram environment; and

means for textually referencing the ~~graphically represented~~ function defined graphically within the model to cause an invocation of the ~~graphically represented~~ function during execution of the model.

71. (Previously Presented) The system of claim 70, wherein the model is represented as a finite state machine.

72. (Previously Presented) The system of claim 71 further comprising:

means for associating the graphically represented function with at least one state or transition within the finite state machine.

73. (Previously Presented) The system of claim 70, wherein the graphically represented function is represented as at least one or a combination of a finite state machine, a state flow diagram, a function flow diagram, and a graphical block diagram model.

74. (Currently Amended) A graphical block diagram modeling system comprising:

a graphical function for use in an executable model, wherein at least subset of commands of the graphical function are defined through a graphical representation; and

a graphical representation of the model including a textual reference of the graphically represented function within the graphical representation of the model to cause an invocation of the graphical function during execution of the model.

75. (Previously Presented) The system of claim 74, wherein the model is represented as a finite state machine.

76. (Previously Presented) The system of claim 75, wherein the finite state machine is a hierarchical finite state machine.

77. (Previously Presented) The system of claim 75, wherein the finite state machine further comprises:

at least one state or transition associated with the graphical function.

78. (Previously Presented) The system of claim 74, wherein the graphical function is represented as at least one or a combination of: a finite state machine, a state flow diagram, a function flow diagram, and a graphical block diagram model.